**PCT** WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 6 : G06F 12/14 — A2

(11) International Publication Number: WO 99/21094

(43) International Publication Date: 29 April 1999 (29.04.99)

(21) International Application Number: PCT/US98/22062

(22) International Filing Date: 19 October 1998 (19.10.98)

(30) Priority Data:
60/063,188 20 October 1997 (20.10.97) US

(71) Applicant: QUICKFLEX, INC. [US/US]; 8409 Cambria Drive, Austin, TX 78717 (US).

(71)(72) Applicant and Inventor: LEDZIUS, Robert, C. [US/US]; 8409 Cambria Drive, Austin, TX 78717 (US).

(74) Agent: SPRINKLE, Steven, R.; Gray Cary Ware & Freidenrich LLP, Suite 1440, 100 Congress Avenue, Austin, TX 78701 (US).

(81) Designated States: BR, CA, CN, JP, European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE).

Published
*Without international search report and to be republished upon receipt of that report.*

(54) Title: RECONFIGURABLE SECURE HARDWARE APPARATUS AND METHOD OF OPERATION

(57) Abstract

A system and method of ensuring that a hardware apparatus in a data-link system can be operated only by an authorized user. The method comprises the steps of assigning a unique identification number to the hardware apparatus, generating at least one security information set for the hardware apparatus which is based on the unique identification number of the hardware apparatus, distributing to an authorized user at least one data string from which the security information set for the hardware apparatus can be derived to functionally enable the hardware apparatus, and inputting the data string into the hardware apparatus to either disable at least one level of functionality of the hardware apparatus if an incorrect security information set is derived from the data string, or to enable at least one level of functionality of the hardware apparatus if a correct security information set is derived from the data string.

*FOR THE PURPOSES OF INFORMATION ONLY*

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

| | |
|---|---|
| AL | Albania |
| AM | Armenia |
| AT | Austria |
| AU | Australia |
| AZ | Azerbaijan |
| BA | Bosnia and Herzegovina |
| BB | Barbados |
| BE | Belgium |
| BF | Burkina Faso |
| BG | Bulgaria |
| BJ | Benin |
| BR | Brazil |
| BY | Belarus |
| CA | Canada |
| CF | Central African Republic |
| CG | Congo |
| CH | Switzerland |
| CI | Côte d'Ivoire |
| CM | Cameroon |
| CN | China |
| CU | Cuba |
| CZ | Czech Republic |
| DE | Germany |
| DK | Denmark |
| EE | Estonia |
| ES | Spain |
| FI | Finland |
| FR | France |
| GA | Gabon |
| GB | United Kingdom |
| GE | Georgia |
| GH | Ghana |
| GN | Guinea |
| GR | Greece |
| HU | Hungary |
| IE | Ireland |
| IL | Israel |
| IS | Iceland |
| IT | Italy |
| JP | Japan |
| KE | Kenya |
| KG | Kyrgyzstan |
| KP | Democratic People's Republic of Korea |
| KR | Republic of Korea |
| KZ | Kazakstan |
| LC | Saint Lucia |
| LI | Liechtenstein |
| LK | Sri Lanka |
| LR | Liberia |
| LS | Lesotho |
| LT | Lithuania |
| LU | Luxembourg |
| LV | Latvia |
| MC | Monaco |
| MD | Republic of Moldova |
| MG | Madagascar |
| MK | The former Yugoslav Republic of Macedonia |
| ML | Mali |
| MN | Mongolia |
| MR | Mauritania |
| MW | Malawi |
| MX | Mexico |
| NE | Niger |
| NL | Netherlands |
| NO | Norway |
| NZ | New Zealand |
| PL | Poland |
| PT | Portugal |
| RO | Romania |
| RU | Russian Federation |
| SD | Sudan |
| SE | Sweden |
| SG | Singapore |
| SI | Slovenia |
| SK | Slovakia |
| SN | Senegal |
| SZ | Swaziland |
| TD | Chad |
| TG | Togo |
| TJ | Tajikistan |
| TM | Turkmenistan |
| TR | Turkey |
| TT | Trinidad and Tobago |
| UA | Ukraine |
| UG | Uganda |
| US | United States of America |
| UZ | Uzbekistan |
| VN | Viet Nam |
| YU | Yugoslavia |
| ZW | Zimbabwe |

# RECONFIGURABLE SECURE HARDWARE APPARATUS AND METHOD OF OPERATION

## TECHNICAL FIELD OF THE INVENTION

This invention relates generally to the field of computer hardware apparatus configuration and more specifically to a system and method of programming and reprogramming a computer hardware apparatus utilizing an encryption key system.

## BACKGROUND OF THE INVENTION

Computer systems today incorporate and interface with a growing number of other devices. Ascertaining with a measure of accuracy that the interface established is between an authorized party or device is often desirable to ensure proper use of computer hardware, software and data.

Solutions have been developed to ensure that data transferred between persons utilizing a computer is only available to authorized parties. One such method of ensuring proper authorization is public key cryptography. Public key cryptography utilizes an encryption key set consisting of two keys. Generally available software can encrypt computer files using either of the keys, such that the computer files are inoperable and unreadable until decrypted. Generally available software can similarly decrypt such encrypted files as long as a user can provide the appropriate complement to the key used to encrypt the files.

Each key in the encryption/decryption key set can be used to encrypt data and its complement key can be used to decrypt data. However, it can be very difficult and very time consuming to determine one key in the set from knowledge of the other. This fact allows a user to make one key of the set public so that others can use this "public key" to encrypt messages prior to sending them to the user. The user keeps the complement, or "private key", secret so that only the user has the means to decrypt messages sent by someone using the public key.

Conversely, a user may use the private key to encrypt a message to be sent to another person. The message can only be decrypted if the recipient has access to the public key. In this way, the recipient can be assured that the author of the message was the holder of the private key. Additionally, if the sender of the message has disclosed the public key to only a small set of recipients, the sender of the message can be reasonably assured that only the intended recipients can decrypt the message, provided that care is taken to limit access to the decryption key.

Software providers can use this same encryption technology to control access to software programs. By encrypting files with one key, and providing the other key on a limited basis, software providers can prevent unauthorized use or copying of their product.

The above solutions, however, only address controlling access to data or a set of software objects. They fail to address security issues

surrounding computer peripherals and their interfaces.

One such peripheral is a Personal Computer Memory Card International Association (PCMCIA) card. These computer cards meet the minimum compliance requirements of the PCMCIA standard (which is hereby incorporated by reference). PCMCIA cards are typically used to add functionality or memory to a personal, portable, or desktop computer (i.e., host computer), as described in the PCMCIA Standard. Many types of PCMCIA cards are available, including input/output (I/O) PCMCIA cards that transfer data between a host computer system and an I/O bus, and data acquisition PCMCIA cards. Typically, data acquisition cards receive and digitize analog information from sensors and temporarily store the information before transferring it to the host computer.

A typical PCMCIA card includes a standard PCMCIA connector connected to a PCMCIA interface circuit through a standard PCMCIA bus. The PCMCIA interface circuit operates according to the standard PCMCIA protocol to send data to and receive data from a host computer. The typical PCMCIA card also may include a PCMCIA card controller that sends data to and receives data from the PCMCIA interface circuit and controls the operation of the functional hardware on the card. For example, if the PCMCIA card is a memory card, then the functional hardware is memory (e.g., a bank of random access memory (RAM) chips or

a hard disk drive) and the PCMCIA card controller controls reading and writing to the memory.

PCMCIA card controllers and interface circuits can be implemented as hardwired logic or as programmable logic (e.g., one or more field programmable gate arrays (FPGAs). The programmable architecture of FPGAs is provided through programmable logic blocks interconnected by a hierarchy of routing resources. The devices are customized by loading programming data into internal static memory cells. FPGA programming data are design-specific data that define the functional operation of the FPGA's internal blocks and their interconnections. Typically, when a PCMCIA card, having the PCMCIA card controller and interface circuit implemented in an FPGA(s), is inserted in an operating (i.e., powered) host computer or is inserted in a powered down host computer that is then powered-up, the FPGA is programmed with FPGA programming data stored in non-volatile memory (e.g., EPROM, EEPROM, Flash memory, etc.) on the PCMCIA card. However, the memory required to store the FPGA programming data generally consumes a measurable area of the PCMCIA card which could be used to provide other functions within the PCMCIA card. Additionally, since these cards are highly pilferable, security protocols should be established to ensure authorized use and programming of these configurable computer hardware devices, especially if the device is intended to be used for and contains key information used to protect data or data access.

Thus there is a need for an improved system and method of ensuring authorized and secure use of a computer hardware apparatus.

There is also a need for an improved system of ensuring authorized and secure programming of configurable computer hardware apparatus.

Additionally, there is a need for an improved system of ensuring authorized and secure programming of re-configurable computer hardware apparatus.

There is also need for an improved system of ensuring authorized and secure communication between re-configurable computer hardware apparatus and host computer systems.

Moreover, there is a need for a system of preventing unauthorized execution of software programs on unauthorized hardware apparatus.

There is also a need for a system that fulfills each of the described needs together in a single system solution.

SUMMARY OF INVENTION

The present invention provides a system and method for reconfiguring a secure hardware apparatus in a data-link system, wherein a data link system comprises a plurality of objects which exchange data, that substantially eliminates or reduces disadvantages and problems associated with previously developed systems and methods for reconfiguring hardware apparatus.

More specifically, the present invention provides a system and method of ensuring that a hardware apparatus in a data-link system can be operated only by an authorized user. The method comprises the steps of assigning a unique number, possibly a serial number, to the hardware apparatus, generating at least one key or key set for the hardware apparatus which is based on the unique serial number of the hardware apparatus, distributing to an authorized user at least one data string from which the key set for the hardware apparatus can be derived to functionally enable the hardware apparatus, and inputting the data string into the hardware apparatus to either disable at least one level of functionality of the hardware apparatus if an incorrect key set is derived from the data string, or to enable at least one level of functionality of the hardware apparatus if a correct key set is derived from the data string.

Additional embodiments of the reconfigurable secure hardware apparatus of the present invention provide a method and system to ensure that the

hardware apparatus can be utilized only by an authorized computer system or group of computer systems and a method and system for ensuring that the reconfigurable secure hardware apparatus can be programmed only by an authorized user utilizing an authorized host computer system.

A further embodiment of the present invention ensures that neither the reconfigurable secure hardware apparatus nor the host computer system will alone have sufficient information to allow a person who has obtained access to only one of either of the host computer system or the reconfigurable secure hardware apparatus to successfully operate the reconfigurable secure hardware apparatus with another unauthorized computer system.

In a still further embodiment of the present invention, a portion of an encrypted code is recorded in a memory location within the reconfigurable secure hardware apparatus and another portion of the encrypted code is recorded in the host computer. This prevents an unauthorized person who obtains access to either the reconfigurable secure hardware apparatus or the host computer database, from discovering more than a portion of the correlating relationship between an account number and the related personal serial number. The present invention also provides for the use of a changeable personal encryption key stored in a memory location. A further embodiment of the present invention can store multiple keys, one for each key set encompassed by the present invention. The storage takes place in

an extension of the standard CIS (Card Information Structure) storage space of the PC-card.

Accordingly, it is an object of this invention to substantially improve the security protocols of a computer hardware apparatus. A technical advantage of the present invention is that it provides a personal identity verification method wherein only part of the information necessary to correlate an account number to characteristic information is available at any one accessible place within the terminal system.

Another technical advantage of the present invention is that it can store an encrypted typed in password for apparatus operation authorization for the entire device key sets, or additional passwords for individual key sets, that may be chosen by the user.

A still further technical advantage of the present invention is the combination of the ability to prevent software piracy and the ability to allow secure user communication via accelerated encryption in a single device.

BRIEF DESCRIPTION OF THE DRAWINGS

A more complete understanding of the present invention and advantages thereof may be acquired by referring to the following description taken in conjunction with the accompanying drawings in which like reference numbers indicate like features and wherein:

FIGURE 1 is a system overview of one embodiment of the reconfigurable secure hardware apparatus of the present invention;

FIGURE 2 is a flow diagram of a method of assigning manufacturing key sets;

FIGURE 3 is a flow diagram of a user registration process according to one embodiment of the present invention;

FIGURE 4 is a flow diagram of a method of registration of software to enable the use of the reconfigurable secure hardware apparatus of the present invention for piracy protection;

FIGURE 5 is a description of a method of machine registration;

FIGURE 6 is a flow diagram of a QARD plug-in procedure according to one embodiment of the present invention;

FIGURE 7 is a description of a method of password protection according to one embodiment of the present invention; and

FIGURE 8 is diagram illustrating an embodiment of the reconfigurable secure hardware apparatus of the present invention.

FIGURE 9 is a diagram illustrating a functional block of the reconfigurable computer hardware apparatus.

DETAILED DESCRIPTION OF THE INVENTION

Several embodiments of the present invention are described in detail below and in the FIGUREs, like numerals being used to refer to like and corresponding parts of the various drawings. However, it should be understood that various changes, substitutions and alterations can be made hereto without departing from the spirit and scope of the invention.

The present invention can be implemented using detachable cards that are operable to be used on various computing devices. For example, a detachable card could be used on a personal computer through a PCMCIA slot. The following description refers to detachable cards used for personal computers (hereafter called "PC-Cards"), but the present invention can be applied to other types of computing devices as well.

One example of a PC-Card that could implement the present invention is a reconfigurable secure hardware apparatus, or Reconfigurable-Computing (RC) card, such as those designed by QUICKFLEX INC., of Austin, Texas. Quick Qard Technology (QQT) is comprised of a family of several PC-Card products that allow different software applications to define the hardware within the PC-Card specific for each application at the time that it is executed. These RC PC-Cards are nicknamed "QUICK QARDS" or "QARDS" and can be used for a variety of standard or custom interfaces, as well as for hardware accelerating software applications. Applications that can benefit

from QQT include personal digital assistant interfaces, PC interfaces, industrial, emulation, video, audio, encryption, computer games, etc.

The security features of the reconfigurable secure hardware apparatus of the present invention described herein can be used for access or piracy protection of third-party software. This third-party software may be comprised of configuration files of hardware apparatus for use within the PC-Cards, or may be general software not targeted to use the reconfigurable computing aspects of the PC-Cards. The security features of the present invention can be implemented as a security system that can be adapted to other types of implementations beyond the QQT products.

Protection of files containing documents, data, executable code, interpretive code or other intellectual property or information which must be protected is achievable using the described security features of the present invention. Protection can be achieved by the use of various sets of public keys. Half of each of the public key sets are stored in the PC-card, which is detachable and thus physically protectable, and half can be stored on the host computer system. Additional security levels of flexible software defined adaptable encryption/decryption algorithms and flexible reconfigurable hardware implementable encryption/decryption algorithms can be implemented within the reconfigurable computing PC-card that allow for customization of the security features.

For purposes of a complete understanding of the scope of the present invention, although reference is made to encryption/decryption algorithms, it should be clear that these are algorithms that are implemented in the reconfigurable computing functional circuits described more completely below for the purpose of hardware accelerating said algorithms. This may be separate from 'check word' algorithms implemented for the purpose of enabling the different levels of security. The later is used to enable the present invention to perform the final functionality of the reconfigurable computing platform for the device.

FIGURE 1 shows a system overview of one embodiment of the present invention that provides protections for information in any form, whether to be kept internal or to be shipped externally, for individual users, groups of users and organizations. FIGURE 1 demonstrates how Quick Qards 1, when used with Anti-Piracy Software 3 and/or Communications & Data Security software 5, result in a Secure Qard system 7. This Secure Qard system 7 can be further used with encryption/decryption key management and authentication software 9 to form an overall Quick Secure system 11.

FIGURE 2 is a flow diagram for a method of assigning unique manufacturing key sets 30 to individual hardware apparatuses 34. Manufacturing key sets 30 can be used to ensure authorized feature enablement of the various features offered by Quick Secure system 11 of FIGURE 1. At step 14 of

FIGURE 4, a unique manufacturing serial number 32 is assigned to each hardware apparatus 34 at manufacture. Unique manufacturing serial number 32 is programmed into the CIS non-volitile memory at manufacturing. The seed value for generating manufacturing key set 30 can be based on unique manufacturing serial number 32 or can be derived by a variety of other methods or algorithms. Step 16 of FIGURE 2 corresponds to manufacturing key set 30 being generated from unique manufacturing serial number 32. Manufacturing key set 30 is the first of the multiple key sets used by the reconfigurable secure hardware apparatus of the present invention and may be referred to as level-zero key set (L0KS) 36 as shown in step 16. Manufacturing key set 30 can be generated and programmed into hardware apparatus 34 at the time it is manufactured. L0KS 36 comprises an encryption key (L0EK) 38 and decryption key (L0DK) 40.

At step 18 of FIGURE 2 the unique manufacturing serial number 32 and the L0DK 40 for a group of manufactured hardware apparatuses can be stored in step 18 in list file 42 for future use. Step 20 creates a registry data file 43 which comprises unique manufacturing serial number 32 and L0EK 38. Registry data file 43 complements list file 42 in that L0DK 40 and L0EK 38 must both be used to enable hardware apparatus 34. Registry data file 43 should be stored in a remote location from hardware apparatus 34 to be accessed by the end user at a later time. This remote location may include a

remote host computer system 86 as shown in step 50 of FIGURE 3 which may be accessed via a communication path such as the internet.

It should be noted that a list of serial numbers and key list for programming into the hardware apparatus could just as easily be generated in advance and given to the manufacturer so that generation of the information is no done on the site of a contract manufacturer who has no need for knowledge of the information that is not to be programmed into the physical hardware apparatus. At step 22, each hardware apparatus 34 is assigned a unique barcode 45 for manufacturing tracking purposes. Barcode 45 may be incorporated into list file 42 and made to correspond to a particular unique serial number 32 and LODK 40 combination. Barcode 45 will ensure that the correct manufacturing serial number 32 and LODK 40 pair are programmed into hardware apparatus 34 during testing of hardware apparatus 34 in step 24. Following testing, hardware apparatus 34 can be packaged in step 26 with a certificate 46 containing the unique manufacturing serial number 32, LODK 40 and barcode 45. This will allow a hardware apparatus 34 designer to ensure that a hardware apparatus 34 registered after purchase was authorized for manufacture by the designer to prevent manufacture of copies of the design by an unauthorized manufacturing house. In step 28 the product is shipped.

During product registration and enablement, which can occur by mail, e-mail, or other electronic

means, unique manufacturing serial number 32 and L0EK 38 (or the manufacturing encryption public key of the set) can be given back to the card designer. This allows the registration information to be checked against registry data file 43 (which comprises a list of approved manufactured PC-Cards) for validity. Also, a card designer can ascertain if a given PC-Card has been previously registered to insure that no un-authorized PC-Card copies with copies of the CIS are being manufactured, as each PC-Card should have a unique code.

FIGURE 3 illustrates one potential registration process for the present invention. In step 50 a communication data path is established between a local computer system 84 and a host computer system 86 wherein hardware apparatus 34 is installed in the local computer system 84. The communication data path may take the form of an internet connection to a "QUICKFLEX" website. A software object operating within host computer system 86 may offer the user a variety of options concerning hardware apparatus 34 wherein the user can select to register the hardware apparatus 34 with the designer in step 52.

Step 54 of FIGURE 3 corresponds to a registration process that can require the user to supply registration information 88 comprising name, email address, information regarding where the hardware apparatus 34 was purchased, and the like. This will allow tracing back to the source of unauthorized hardware apparatus 34 manufacturing. Most importantly, the user will be prompted to supply

either a certificate number which corresponds to barcode 45 of FIGURE 2 or to unique manufacturing serial number 32 and L0DK 40.

In step 56 a check is performed to determine if hardware apparatus 34 support software 90 is installed on local computer system 84. Based on the results of this check a decision is made in step 58 to either download and install the necessary software support 90 at step 60 or to proceed to step 62. Step 62 provides for establishing a secure link 92 between hardware apparatus 34 and host computer system 86 if software 90 is present on local computer system 84. Secure link 92 provides for the transfer of unique manufacturing serial number 32 and L0DK 40 from a programmed memory location within hardware apparatus 34 to host computer system 86.

At step 64 of FIGURE 3 the user is required to manually enter barcode 45 or the unique manufacturing serial number 32 and L0DK 40 pair. At step 66, a verification is performed on manually entered barcode 45 or unique manufacturing serial number 32 and L0DK 40 pair against a copy of unique manufacturing serial number 32 and L0DK 40 transferred from a programmed memory location within hardware apparatus 34. If the verification fails the user is prompted with an error message at steep 68 to return to registration data entry process step 54. If the verification is successful, additional verifications are made in step 70 to verify that unique manufacturing serial number 32 is contained within registry data file 43 and in step 72 to verify that hardware apparatus 34 has not

been previously registered. Any problems associated with these verifications force the user to contact the designers concerning the registration error as shown in step 74 so that the problem can be identified and resolved.

Registry data file 45 is updated in step 76 of FIGURE 3 to include the information associated with unique manufacturing serial number provided in step 54. To further prevent misappropriation or unauthorized use of hardware apparatus 34, step 78 generates a new L0KS 36 comprising a new L0DK 40 and a new L0EK 38. At step 80, a secure link is again established between hardware apparatus 34 and host computer system 86 allowing host computer system 86 to reprogram new L0DK 40 into a memory location of hardware apparatus 34. A new L0EK 38 is also downloaded to the user that can be recorded on certificate 46 or programmed directly into a memory location of hardware apparatus 34. Registry data file 43 is also updated with the new L0KS 36 in step 82.

An important technical advantage associated with the present invention allows a software vendor to prevent unauthorized use of its proprietary software. Software or configuration file vendors or authors can use the L0KS 36 encryption key for providing an access code for licensing or allowing their Intellectual Property (IP) contained in virtual hardware objects for the RC system to be accessed by one and only one Secure Qard user. They may also limit the time span in which their IP is accessible

or limit the number of times their IP is accessible to the user with other security provisions. Vendors can also use an on-line card designer's public key listing of users, provided that users allow this at registration time, to verify that a given user is registered for utilizing the secure authorization code.

FIGURE 4 is a flow diagram of a method of registration of software to enable the use of the current invention for software piracy protection. The user establishes a communication path in step 90 via the internet or other means between local computer system 84 containing hardware apparatus 34 and a vendor (host) computer system 86. In step 92, the user is prompted by software vendor computer system 86 to select an option allowing the user to register a software application 434. Step 94 requires the user to supply registration information 120 which may be comprised of name, email address, information regarding where software application 434 was purchased, the unique software registration number 124 and the like to vendor computer system 86. Vendor computer system 86 can access unique manufacturing serial number 32 of hardware apparatus 34 directly from a memory location within hardware apparatus 34 as shown in step 96. In step 98, vendor computer system 86 establishes a communication path to software registry database 122. Software registry database 122 may be contained in a third computer system 424 and can comprise a website, such as QUICKFLEX INC.'s registry website. At step 102,

unique software registration number 124 is submitted to the third computer system 424 software registry database 122.

In step 104 unique software registration number 124 is compared to the entries in software registry database 122 to determine if it is a valid unique software registration number 124. If unique software registration number 124 is not valid, an error message will be generated at step 106 that is echoed by vendor computer system 86 to the user in step 108. If unique software registration number 124 is valid, vendor computer system 86 can supply a software authorization code 126 in step 110 to be sent to third computer system 424.

At step 112, third computer system 424 generates a software run code 128 for hardware apparatus 34. Software run code 128 is transmitted to vendor computer system 86 which echoes it to hardware apparatus 34. Software run code 128 can allow the vendor software to be installed on the local computer system or the vendor software can verify the presence of software run code 128 on hardware apparatus 34 before executing the vendor software. Both the authorized installation of the vendor software on a given local computer system 32 and the authorized execution of the vendor software are thus ensured.

At step 116, a counter 750 counts upward by one for each software run code 128 sent to vendor computer system 86 to account for possible royalty payments. In step 118, software vendor computer

system 86 sets a license in place for the user to use the software.

An important technical advantage associated with the present invention allows hardware apparatus 34 to be operated not only by a specific user but also only on a specific local computer system 84. Local computer system 84 may comprise a group of individual computers. FIGURE 5 is a method of ensuring that hardware apparatus 34 is utilized only by authorized local computer systems 84. FIGURE 5 uses the QUICK QARD system of FIGURE 1 for illustrative purposes, but any secure hardware apparatus, reconfigurable or not, of the present invention can be used instead. Support software 90 of hardware apparatus 34 is installed and executed on the local computer system 84. A communication path is established between hardware apparatus 34 and local computer system 84. A verification is made to determine if hardware apparatus 34 is password protected. If so, a valid password must be supplied before proceeding. Support software 90 will verify if unique manufacturing serial number 32 of hardware apparatus 34 is in a registry list 130 maintained on local computer system 84. If unique manufacturing serial number 32 of hardware apparatus 34 is not in registry list 130, the user must register hardware apparatus 34. The list of authorized host ID's permutated with the unique serial number or key set information could also be stored within the EEPROM memory of the apparatus for allowing apparatus enabling on a

particular machine as well. In this case the host ID must be registered with the apparatus.

One such registration method is described in FIGURE 5. Both software and hardware must be installed and registered for each computer in local computer system 84. This feature allows a PC-Card to be locked for use on one or a group of machines. Registration with a card designer can insure that if a L0EK 38 certificate or a password is forgotten or lost, the PC-Card can be reprogrammed to erase the password and create a new manufacturing L0KS 36 for the PC-Card, or to program a recoverable SN & Keyset on the registered user's request. This insures that no encrypted data may be compromised.

Memory space in the CIS memory device can be made available for the purpose of holding an encrypted password defined by the user. In such fashion, the present invention can insure that the PC-Card is only used by that user. This memory space can be left cleared at test (all zero's) and can be enabled for password protection if the purchaser decides to activate that feature. Users may define any password they wish and the entry can then be encrypted using the L0EK 38 that resides on the machine during the initial setup of the PC-Card after purchase. When password checks are made, the encrypted password programmed into the PC-Card can be decrypted using L0DK 40 and can be checked against the typed in value.

PC-Cards implementing the current invention can also be configured to only execute on a specific

machine or group of machines with the use of passwords, thus making the PC-Card hardware of little use in the event it is stolen. FIGURE 5 provides a detailed description of one method of machine registration.

FIGURE 6 is a flow diagram of a QARD plug-in procedure according to one embodiment of the present invention. In step 150, hardware apparatus 34 is installed in a local computer system 84. A check is performed in step 152 to determine the presence in local computer system 84 of the necessary support software 90. If support software 90 is not present, the QARD plug-in procedure terminates and the support software 90 must be installed before resuming with the QARD plug-in procedure.

Hardware apparatus 34 may be protected by a password and step 154 tests to determine if password protection is enabled. If password protection is enabled, the password must be provided in step 156. At step 158 the provided password is encrypted using LOEK 38. If the provided password matches the password stored in a memory location on hardware apparatus 34, then at step 160 the plug in procedure is allowed to proceed. Furthermore, hardware apparatus 34 may be protected by a verification step, to verify authorization by local computer system 84, requiring the input of a match to unique serial number 174 provided by local computer system 84 (such as by a hard disk drive serial number). Hardware apparatus 34's use is not permitted unless the unique serial number 174 inputted at step 162 matches the

unique serial number 174 stored in local computer system 84. Step 164 determines if the inputted serial number 174 is a match. If it is, than the QARD plug-in procedure is complete.

Unique serial number 174 is generated during the user registration process as described above for FIGURE 3. It can be stored either in hardware apparatus 34 memory or in local computer system 84 memory. If unique serial number 174 does not match at step 164, then the user registration process of FIGURE 3 must be performed in steps 168-172 of FIGURE 6 to complete the QARD plug in procedure.

FIGURE 7 is a description of a method of password protection according to one embodiment of the present invention which additionally is illustrated as part of the flow diagram presented in FIGURE 6. Steps 154-160 of FIGURE 6 correspond to this method of password protection.

FIGURE 8 illustrates another embodiment of the reconfigurable secure hardware apparatus of the present invention. Reconfigurable hardware apparatus 100 interfaces with a host computer system 200 or with another hardware apparatus. Reconfigurable hardware apparatus 100 may be divided into three modules, a configuration control module 300, a configuration status module 400 and a functional module 500. Host computer system 200 interfaces with reconfigurable hardware apparatus 100 by way of data input/output bus 202. Input/output bus 202 is shown accessing four control data registers, 304a, 304b, 304c and 304d, inside configuration control module

300 and two status data registers, 306a and 306b, inside configuration status module 400. Control data registers 304a-304d provide a temporary storage location for data transmitted or received from data input/output bus 202. While this embodiment of the reconfigurable secure hardware apparatus of the present invention has been described with four data registers, it can have more or less registers, as needed.

Code Generator (CG) 310 accepts input data from data register 304b to generate a check data word. The check data word generated by CG 310 can be LOKS 36. Multiple generated check words can be generated for different security features for enabling the separate security features of the invention. Since the features described are security related, a process or algorithm for generating the check data words should be kept as a trade secret for an organization producing reconfigurable hardware apparatus 100. The process chosen should yield as output check data words that are not easily determined from the input data to the process, which could be comprised of manufacturing serial number 32, and should have properties that output a pseudo-random sequence that is sufficient in length to not easily be guessed by trial and error.

The embodiment of the reconfigurable secure hardware apparatus of the present invention described herein is only one of many possible implementations and is provided for illustrative purposes only. The focus of this embodiment of the present invention is

the way in which CG 310 is used to realize the security features described. The check data words outputted by CG 310 can be checked with code comparator (CC) 312 against an input check value stored in data registers 304c and 304d, which together comprise the Code Check Register (CCR) 314. The input check value stored in CCR 314 can be comprised of L0EK 38 and L0DK 40 and can also be user inputted. CCR 314 can be a register having a length equal to the length of the CG 310 check data word output and can be written to allow an authorization check of the reconfigurable hardware apparatus 100 feature being used. Longer check data words may require multiple CCRs 314 if they extend beyond the host computer system 200 data bus width. The values written to CCR 314 may be provided in several different manners depending on what feature of reconfigurable hardware apparatus 100 is being authorized.

CC 312 performs a bit-by-bit check of the CG 310 check data word output and the entered CCR 314 value to determine if the feature authorization check passes or fails. If the feature authorization check passes, CC 312 generates a high digital bit output (a digital "1") and forwards it to configuration and control gates 318, which is comprised of a plurality of "AND" logic gates 700 corresponding to the plurality of features of reconfigurable hardware apparatus 100. These features include, but are not limited to, product enable check 319, HDD ID enable

Check 320, Flash Write enable 322, password enable 324 and Configuration File Vendor Enable 326.

Control register 316 receives an input from host computer system 200 through data registers 304a and 304b to select one or more of the features of reconfigurable hardware apparatus 100. Based on the input received from data registers 304a and 304b, control register 316 will generate a high digital bit output ("1") as an input for the selected features and a low digital bit output ("0") as an input for all the other features. The AND gates 700 for the selected features will therefore have two high digital bit inputs and will output a high digital bit as an input to their corresponding pull-down resistors 350 in configuration status module 400, thereby allowing access to the corresponding feature in functionality circuit 500 as directed by host computer system 200. The pull down resistors are necessary if it is possible that the Security Login Module 300 disappears due to the module being implemented within the FPGA of the RC hardware apparatus 100.

In this manner, configuration status module 400 can be instructed to reprogram and enable the various features of functionality circuit 500 depending on which features are so selected. Functional Module 500 may receive virtual hardware objects for performing applications specific tasks within the reconfigurable computing hardware apparatus FPGA. Additionally, status data registers 306a and 306b can interface with host computer system 200 through data

input/output bus 202 to communicate the configuration of functionality circuit 500 to host computer system 200.

Representative input check value sources for the various features of reconfigurable secure hardware apparatus 100 for the various embodiments of the present invention are shown in the following table:

| **Security Feature:** | **CCR 14 Source:** |
|---|---|
| Product Operation Enable | 1st portion of Product Enable Certificate Code (After 1st authorization, the code is programmed into FLASH memory of the hardware apparatus for automated driver access) |
| New Machine Operation Registration | 2nd portion of Product Enable Certificate Code (never programmed into FLASH memory of the hardware apparatus, always required to be typed in) |
| Machine Operation Enable | Read by driver from a list of HDD codes entered in FLASH memory of the hardware apparatus and each checked until a match is found or the list is exhausted. The HDD ID's are obtainable by reading this list, as the values are the results of CG's of |

| | |
|---|---|
| | previously authorized machines. |
| Password Operation Enable | Read by the driver from the FLASH attribute memory and programmed into the CCR. The password is not obtainable from looking at the CG result of the password. |
| Anti-Piracy Operation Enable (this feature may contain a vendor defined specific CG different from what is used in QQT features) | Written by application software program |

FIGURE 9 provides a functional block diagram 400 of the reconfigurable computing hardware apparatus used to illustrate the reconfigurable computing operations that the present invention makes possible. EEPROM 410 provides CIS memory, key memory, and password storage functions. Programmable Logic Device 420, which may be an application specific integrated circuit provides interface/configuration/and status register functions. In addition, the security feature circuit (block 300, FIGURE 8), which may be within FPGA 420 or within PLD configuration register 430, provides the necessary implementation for these functions. Field programmable gate array 420

implements security feature circuit functions of the present embodiment. Either a programmable logic device or field programmable gate array 420 may make possible the reconfigurable computing functional circuits. Virtual hardware objects 500 of FIGURE 8, attentively, may also provide these functions. Host bus interface socket 440 includes a 68-pin PCMCIA connector. Other components of FIGURE 9, including the various generic items such as oscillators 450, expansion connectors 460 and 470, RAM 480, or other features provide the ability to apply the reconfigurable computing to a desired application.

The following paragraphs provide a description of several additional features and terms for the different embodiments of the reconfigurable secure hardware apparatus of the present invention and their operation.

Level-One Key set (L1KS): User Public Key

A L1KS space can be provided for a user to generate and define a key set specific for that user which is not registered with the card designer and is kept secret by the user. The L1KS can be stored just like L0KS 36. Password space for a level-one password (L1PW) can also be allocated in the CIS and can execute in the same manner as the level-zero password (L0PW).

The L1KS can be generated by the user and thus there are no guarantees that the code is unique. The bit length can be long enough, however, to insure

that it is improbable that the key set is in use by another user. The bit length of this key can differ in length from LOKS 36. The user public key is a secure key set and may be changed by the user over time.

This key is useful when the information is intended only for the user. Even so, however, a further advantage of the present invention is that several PC-Cards may be programmed with the same LiKS for project sharing access. The key set for the group could be common to all PC-Cards used by the group. An example of usage of this key is for encrypting and decrypting information regarding a common project where access is required by multiple project members. Members of the group may be local or remote and may securely exchange data utilizing this key.

Additional Key sets

The present invention contemplates that the number of key sets can be expanded beyond the two sets defined in the above sections.

File Header Information

This section describes how one embodiment of the present invention uses header information of an encrypted file which utilizes the security features of the present invention.

A file header can contain the following information in addition to the normal file header information normally found in files for a particular operating system. The present embodiment could wrap this additional header information around the information indicated.

The following is a description of an embodiment of the present invention implemented USING a QQT card of QUICKFLEX INC. with two encryption levels.

| | |
|---|---|
| QQTSL: (0,1) | Quick PC-Card Technology Security Level 0 or 1 |
| KIND | Kind of file |
| AN | The name of the encryption algorithm used in encrypting the file. |
| LnEK | Level-n Encryption Key used for encrypting the file |
| EMD | Encrypted Message Data |
| EFD | Encrypted File Data |

QQTSL (1 or 1): QUICK PC-Card Technology Security Level 0 or 1:

This information can indicate the security level of the key used for the encryption process. QQTSL0 and QQTSL1 correspond to the L0KS and L1KS, respectively, used in the PC-Card.

KIND: Kind of File

This information indicates one of the following kinds of files:

| | |
|---|---|
| QQT: model | Quick PC-Card Technology configuration file for specified PC-Card model. |
| EXE: type | Executable file for defined type of machine and operating system. |
| OTHER | Other kind of file. |

The QQT PCMCIA driver during a configuration load can automatically decrypt QQT files using the specified algorithm. During the load process by an application program, information in the file for a window message can be displayed indicating information the author wishes to be displayed and the user must respond to the window to continue execution. Notices such as "QQT Module: name is the property of Company XYZ and may not be sold or distributed without the prior written consent of Company XYZ". This enables the author to freely distribute hardware apparatuses for PC-Cards, allowing possible developers who may be interested in licensing the hardware apparatuss the ability to evaluate the work prior to agreeing to license rights to the work. Encrypted configuration files may be encrypted for use only with certain PC-Cards to protect against mass unauthorized distribution of the intellectual property. Generation of unique encrypted hardware apparatuss for target evaluation PC-Cards can be done automatically and transparently through a web site. The requesting party can be

required to have a PC-Card and register the PC-Card at the site in order to build an encrypted configuration file of the hardware apparatus for evaluation purposes.

AN: Algorithm Name

Encryption algorithms used to encrypt or decrypt files can be changed over time. Groups of users or a software vendor may develop their own custom algorithm. Algorithms may be executed as software or as hardware within a RC PC-Card, provided the RC PC-Card has enough gate capacity to execute the defined algorithm in hardware. For example, the QQT driver has a default algorithm built into it that is executed in software as data is passed through the PC-Card for configuration file protection purposes.

Algorithms utilizing run time authorization codes, date expiration codes, or other access limits may utilize additional external information other than that found in the encrypted file that needs to be supplied by the source of the encrypted data for access.

Level (0 or 1) Encryption Key

This information is the Public Key Encryption Key used for encrypting the file. This key can be originally supplied by a receiver and made public. The key length can be derived from the QQT SECURE FILE indication on the first line. It is included in the file so that an easy and fast determination of the target destination for the data can be verified.

EMD: Encrypted Message Data

The EMD contains information regarding the contents of the file that may be decrypted and looked at quickly without decrypting the entire EFD. For example, the EMD for a QQT configuration file is displayed in a window whenever the file is loaded. Certain algorithms may also utilize the EMD to transmit an additional encrypted key for decrypting the EFD with a non -public key algorithm. In other words, the security system may use public key to secretly transmit a separate secure key.

EFD: Encrypted File Data

The EFD contains the encrypted file data including original operating system header information.

Purchased Authorizations of Software

An additional technical advantage the present invention is to allow flexibility for software distributors. For example, the software distributor could freely distribute software or provide the

software in a freely downloadable format to the public, but in order for the software to be executed, a valid authorization code must be present. The software vendor could create an authorization code that corresponds to a specific PC-Card encompassing the present invention. Just as configuration files for the PC-Cards can be obtained via a web page, authorization codes to run software can be purchased via a web page. Each user would need only one PC-Card to allow authorization of running any software utilizing the piracy aspects of the invention. Each software vendor may also define their own algorithms for protecting their software using the PC-Cards. A machine could run the software as long as the authorization codes for the particular QARD used in this system is present to validate the execution of the software. A further embodiment of the invention would allow a database of authorized QARD users to be made available to software vendors.

A further embodiment of the invention could use RC aspects of a PC-Card in order to allow the user to define hardware encryption / decryption algorithms that could be changed over time.

Although the present invention has been described in detail herein with reference to the illustrative embodiments, it should be understood that the description is by way of example only and is not to be construed in a limiting sense. It is to be further understood, therefore, that numerous changes in the details of the embodiments of the invention and additional embodiments of the invention will be

apparent to, and may be made by, persons of ordinary skill in the art having reference to this description. It is contemplated that all such changes and additional embodiments are within the spirit and true scope of the invention as claimed below.

WHAT IS CLAIMED IS:

1. A reconfigurable computing system for incorporating into a personal computer portable removable interface, comprising:

reconfigurable computing circuitry comprising flexibly configurable circuitry for enabling a plurality of security features;

memory circuitry associated with said reconfigurable computing circuitry for storing a plurality of personal security information, and

said reconfigurable computing circuitry and said memory circuitry packaged for portable association along with a personal computer.

2. The reconfigurable computing system of Claim 1, further comprising circuitry for changing data protection cryptography algorithmic hardware for accelerating the operation of hardware implementation security algorithms associated with said reconfigurable computing circuitry.

3. The reconfigurable computing system of Claim 1, wherein said personal security information comprises a public key set.

4. The reconfigurable computing system of Claim 1, wherein said personal security information comprises a private key set.

5. The reconfigurable computing system of Claim 1, further comprising:

a data input/output system to allow a transfer of data between the reconfigurable secure hardware apparatus and a first host computer system;

a plurality of data registers to accept at least one data input from the data input/output system;

a code generator to accept at least one data input from at least one data register and generate an output code;

a code comparator to compare an authorization code stored in at least one data register to the output code of the code generator and send a signal representing whether the authorization code and the output code are identical;

a control register which specifies to a plurality of logic circuits which functions of the reconfigurable secure hardware apparatus are to be examined for enablement wherein the plurality of logic circuits provide at least one signal to a configuration register based on the input of the code comparator and control register; and

at least one functionality circuit operably connected to the configuration register wherein the functionality of the functionality is specified by the configuration register.

6. The reconfigurable secure hardware apparatus of Claim 5, wherein the at least one functionality circuit further comprises at least one external input/output bus connector.

7. The reconfigurable secure hardware apparatus of Claim 5, wherein reconfigurable secure hardware apparatus comprises a PCMCIA card.

8. The reconfigurable secure hardware apparatus of Claim 5, wherein at least one data register is used as a code check register to provide an input to the code comparator.

9. A method for reconfigurably computing security features for a personal computer modem card interface for ensuring hardware apparatus operation in a data-link system only by an authorized user, comprising:

enabling a plurality of security features using a reconfigurable computing circuitry comprising flexibly configurable circuitry;

storing a plurality of personal security information sets using a memory circuitry associated with said reconfigurable computing circuitry; and

packaging said reconfigurable computing circuitry and said memory circuitry for association within a personal computer.

10. The method of Claim 9, further comprising the steps of changing data protection cryptography

algorithmic hardware for accelerating the operation of hardware implementation security algorithms associated with said reconfigurable computing circuitry.

11. The method of Claim 9, further comprising the steps of:

assigning a unique identification number to the hardware apparatus;

generating a first level-zero security information set for the hardware apparatus wherein the first level-zero security information set is based on the unique identification number of the hardware apparatus;

distributing to an authorized user at least one data string from which can be derived the first level-zero security information set for the hardware apparatus to functionally enable the hardware apparatus; and

inputting into the hardware apparatus the at least one data string wherein at least one level of functionality of the hardware apparatus is disabled if an incorrect first level-zero security information set is derived from the data string and at least one level of functionality of the hardware apparatus is enabled if a correct first level-zero security information set is derived from the data string.

12. The method of Claim 11, further comprising the steps of:

forming the first level-zero security information set with a first encryption code data string and a first decryption code data string;

programming the unique identification number and the first level-zero security information set into a memory location of the hardware apparatus;

distributing the unique identification number and the first decryption code to the authorized user of the hardware apparatus;

separately distributing the first encryption code to the authorized user;

entering the unique identification number, the first encryption code, and the first decryption code into at least one data register of the hardware apparatus;

verifying the unique identification number entered into the at least one data register of the hardware apparatus matches the unique identification number programmed into the memory location of the hardware apparatus disabling at least one level of functionality of the hardware apparatus if the entered unique identification number does not match the programmed unique identification number; and

combining the entered first encryption code and the entered first decryption code to form the data string through the use of an algorithm.

13. The method of Claim 11, wherein the algorithm utilized to combine the first encryption code and the first decryption code is an adaptable encryption/decryption algorithm.

14. The method of Claim 11, further comprising registering the hardware apparatus, comprising the steps of:

establishing a communication path from the hardware apparatus to a host computer system;

choosing an option to register the hardware apparatus from a plurality of options offered by a software object operating on the host computer system;

supplying registration information which identifies the authorized user and the unique identification number for the hardware apparatus;

determining if a software application to operate the hardware apparatus is present on an authorized user's computer;

transferring the software application to the authorized user's computer if the software application is not present on the authorized user's computer;

establishing a secure link utilizing the software application from the memory location of the hardware apparatus to the host computer system;

transferring the unique identification number for the hardware apparatus and the first decryption code to an encryption/decryption database inside the host computer system;

verifying that the hardware apparatus has not been previously registered with a registration database located on the host computer system, and wherein at least one level of functionality will be

disabled if the hardware apparatus has been previously registered;

updating the registration database located on the host computer with the registration information; and

transferring the first encryption code to the hardware apparatus.

15. The method of Claim 11, wherein the step of separately distributing the first encryption code to the authorized user is accomplished after the authorized user has registered the hardware apparatus.

16. The method of Claim 11, wherein the step of establishing a communication path from the hardware apparatus to a host computer system comprises utilizing a local computer system to navigate to an internet web site operated by a host computer system.

17. The method of Claim 11, wherein the step of establishing a communication path from the hardware apparatus to the host computer system comprises a secure communication path utilizing an internet connection to the host computer system.

18. The method of Claim 11, wherein registration of the hardware apparatus further comprises the steps of:

generating a second level-zero security information set from a second data string that is

different from the first level-zero security information set generated using the first data string;

generating a second encryption code and a second decryption code based on the second level-zero security information set;

updating the encryption/decryption database inside the host computer system with the second level-zero security information set, the second encryption code and the second decryption code associated with the unique identification number for the hardware apparatus;

erasing the first level-zero security information set from the memory location of the hardware apparatus;

programming the hardware apparatus with the second level-zero security information set wherein the second level-zero security information set may be used in place of the first level-zero security information set for any later registration events; and

distributing to an authorized user a second decryption code comprising at least one data string from which can be derived the second level-zero security information set for the hardware apparatus which may be used in place of the first level-zero security information set for any later registration events.

19. The method of Claim 11, wherein the communication path between the hardware apparatus and

the host computer system is contained within a secure connection.

20. A method of ensuring that a hardware apparatus in a data linked system can be operated only with an authorized local computer system comprising the steps of:

assigning a first unique identification number to the hardware apparatus;

assigning a second unique identification number to the authorized local computer system that can be accessed by the hardware apparatus;

generating a first level-zero security information set for the hardware apparatus which is formed from a first decryption code and a first encryption code;

distributing to the authorized user the first decryption code from which can be derived the first level-zero security information set for the hardware apparatus which is based on the first unique identification number to functionally enable the hardware apparatus;

programming the first unique identification number, the first decryption code, a copy of the second unique identification number and the first level-zero security information set into a memory location of the hardware apparatus;

inputting into the hardware apparatus the first encryption code which combines with the first decryption code to form a data string corresponding to the first level-zero security information set

wherein at least one level of functionality of the hardware apparatus is disabled if the data string does not exactly match the first level-zero security information set;

verifying that the second unique identification number of the authorized local computer system exactly matches the copy of the second unique identification number programmed into the memory location of the hardware apparatus wherein at least one level of functionality of the hardware apparatus is disabled if the second unique identification number of the authorized local computer system that can be accessed by the hardware apparatus does not exactly match the copy of the second unique identification number programmed into the memory location of the hardware apparatus.

21. The method of Claim 20, further comprising the steps of:

separately distributing the first encryption code to the authorized user;

entering the first unique identification number, the first encryption code, and the first decryption code into the hardware apparatus; and

combining the first encryption code and the first decryption code to form the data string corresponding to the first level-zero security information set through the use of an algorithm.

22. The method of Claim 20, wherein the algorithm utilized to combine the first encryption

code and the first decryption code is an adaptable encryption/ decryption algorithm.

23. The method of Claim 20, wherein the step of verifying the second unique identification number of the authorized local computer system further comprises verifying that the second unique identification number that can be accessed by the hardware apparatus exactly matches one of a group of second unique identification codes wherein the group of second unique identification codes corresponds to a group of local computer systems wherein the group of second unique identification codes are programmed into a memory location of the hardware apparatus wherein at least one level of functionality of the hardware apparatus is disabled if the second unique identification number of the authorized local computer system that can be accessed by the hardware apparatus does not exactly match one of the group of second unique identification codes.

24. The method of Claim 20, wherein the hardware apparatus is a reconfigurable secure hardware apparatus.

25. A method of ensuring that a software application cannot be installed or executed on an unauthorized local computer system comprising the steps of:

establishing a data path between a hardware apparatus and a host computer system;

choosing an option to register the software application from a plurality of options offered by a software object operated on the host computer system;

transferring a unique identification number for the software application collected during the option to register the software application from the host computer system to a software vendor computer system containing a software registry database;

verifying the unique identification number entered for the software application matches information contained in the software registry database; and

providing a software run code from the software vendor computer system to the local computer system via the host computer system wherein at least one level of functionality will be disabled if the unique identification number does not match information contained in the software registry database.

26. The method of Claim 25, wherein the step of establishing a data path between the hardware apparatus and the host computer system comprises utilizing a local computer system which contains the hardware apparatus to navigate to an internet web site operated on the host computer system.

27. The method of Claim 25, wherein the data path between the hardware apparatus and the host computer system is contained within a secure connection.

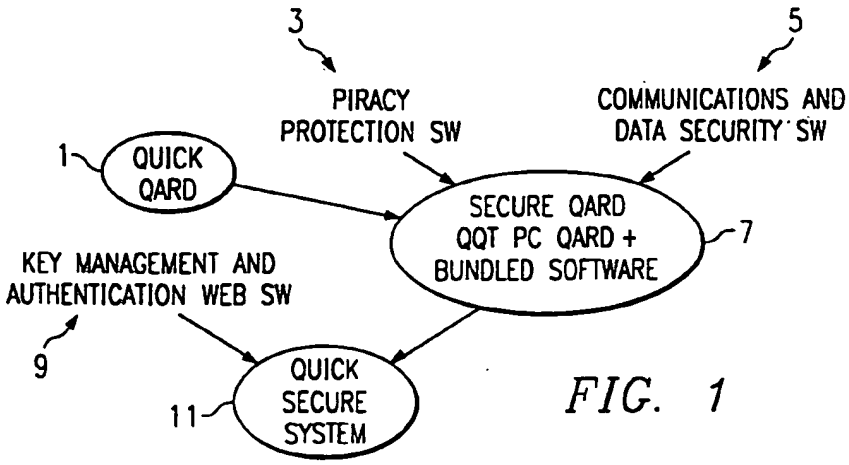

3
PIRACY PROTECTION SW
5
COMMUNICATIONS AND DATA SECURITY SW
1
QUICK QARD
SECURE QARD QOT PC QARD + BUNDLED SOFTWARE
7
KEY MANAGEMENT AND AUTHENTICATION WEB SW
9
11
QUICK SECURE SYSTEM

FIG. 1

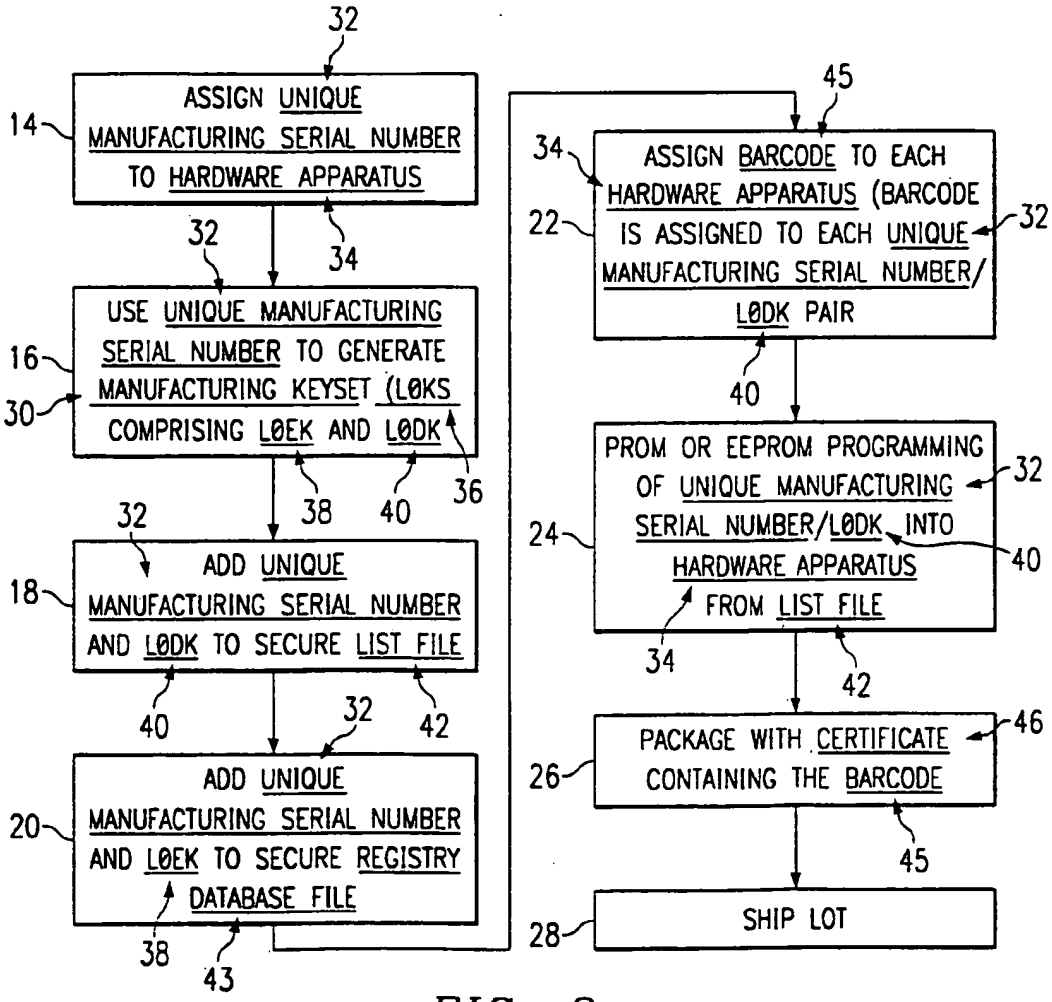

32
ASSIGN UNIQUE MANUFACTURING SERIAL NUMBER TO HARDWARE APPARATUS
14
32
34
USE UNIQUE MANUFACTURING SERIAL NUMBER TO GENERATE MANUFACTURING KEYSET (L0KS COMPRISING L0EK AND L0DK
16
30
38
40
36
32
ADD UNIQUE MANUFACTURING SERIAL NUMBER AND L0DK TO SECURE LIST FILE
18
40
32
42
ADD UNIQUE MANUFACTURING SERIAL NUMBER AND L0EK TO SECURE REGISTRY DATABASE FILE
20
38
43
45
ASSIGN BARCODE TO EACH HARDWARE APPARATUS (BARCODE IS ASSIGNED TO EACH UNIQUE MANUFACTURING SERIAL NUMBER/ L0DK PAIR
34
22
32
40
PROM OR EEPROM PROGRAMMING OF UNIQUE MANUFACTURING SERIAL NUMBER/L0DK INTO HARDWARE APPARATUS FROM LIST FILE
24
32
40
34
42
PACKAGE WITH CERTIFICATE CONTAINING THE BARCODE
26
46
45
SHIP LOT
28

FIG. 2

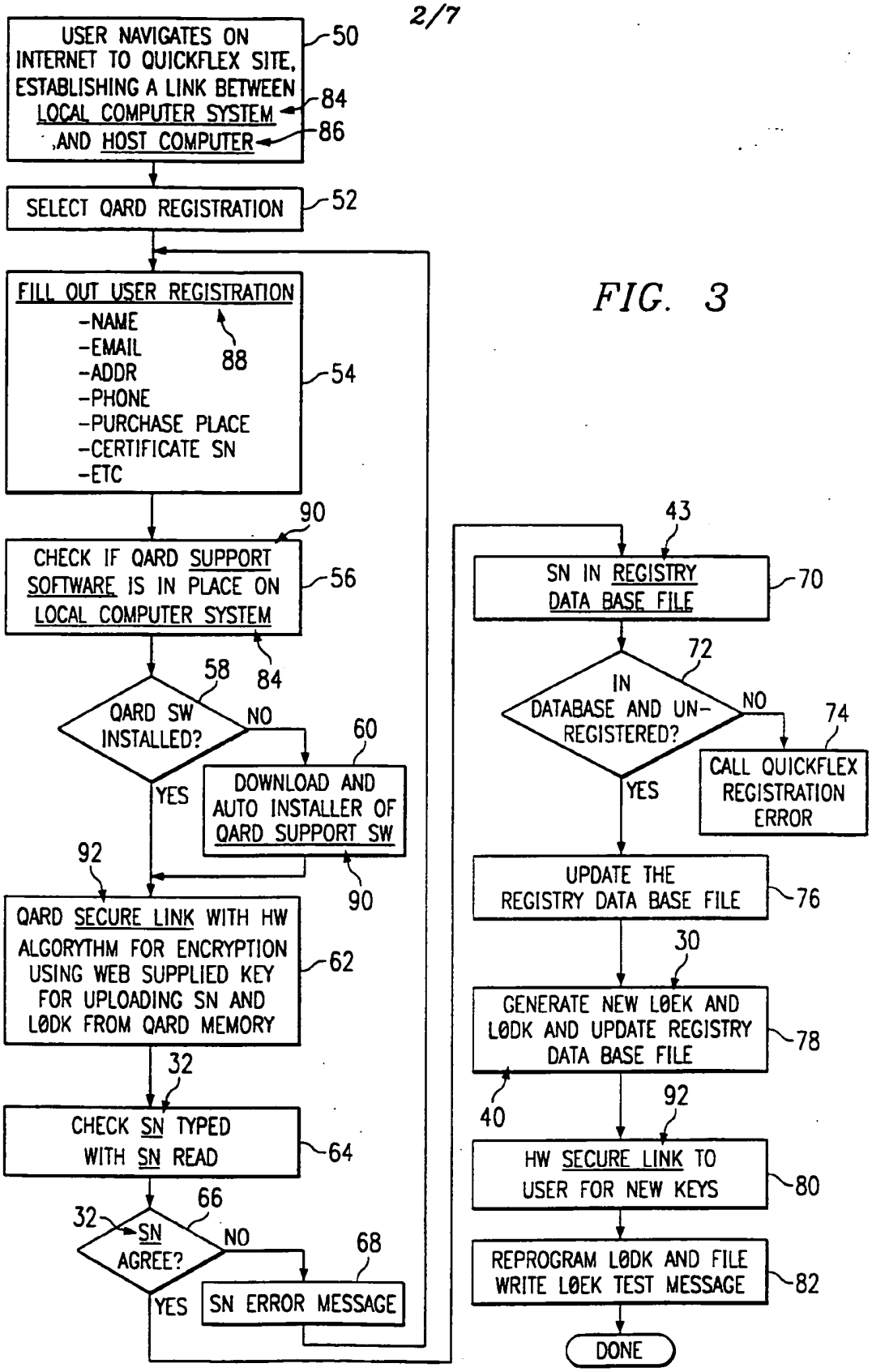

USER NAVIGATES ON INTERNET TO QUICKFLEX SITE, ESTABLISHING A LINK BETWEEN LOCAL COMPUTER SYSTEM AND HOST COMPUTER
50
84
86
SELECT QARD REGISTRATION
52
FILL OUT USER REGISTRATION
-NAME
-EMAIL
-ADDR
-PHONE
-PURCHASE PLACE
-CERTIFICATE SN
-ETC
88
54
CHECK IF QARD SUPPORT SOFTWARE IS IN PLACE ON LOCAL COMPUTER SYSTEM
90
56
84
QARD SW INSTALLED?
58
NO
YES
DOWNLOAD AND AUTO INSTALLER OF QARD SUPPORT SW
60
90
92
QARD SECURE LINK WITH HW ALGORYTHM FOR ENCRYPTION USING WEB SUPPLIED KEY FOR UPLOADING SN AND LØDK FROM QARD MEMORY
62
32
CHECK SN TYPED WITH SN READ
64
66
32
SN AGREE?
NO
YES
68
SN ERROR MESSAGE
43
SN IN REGISTRY DATA BASE FILE
70
72
IN DATABASE AND UN-REGISTERED?
NO
YES
74
CALL QUICKFLEX REGISTRATION ERROR
UPDATE THE REGISTRY DATA BASE FILE
76
30
GENERATE NEW LØEK AND LØDK AND UPDATE REGISTRY DATA BASE FILE
78
40
92
HW SECURE LINK TO USER FOR NEW KEYS
80
REPROGRAM LØDK AND FILE WRITE LØEK TEST MESSAGE
82
DONE

FIG. 3

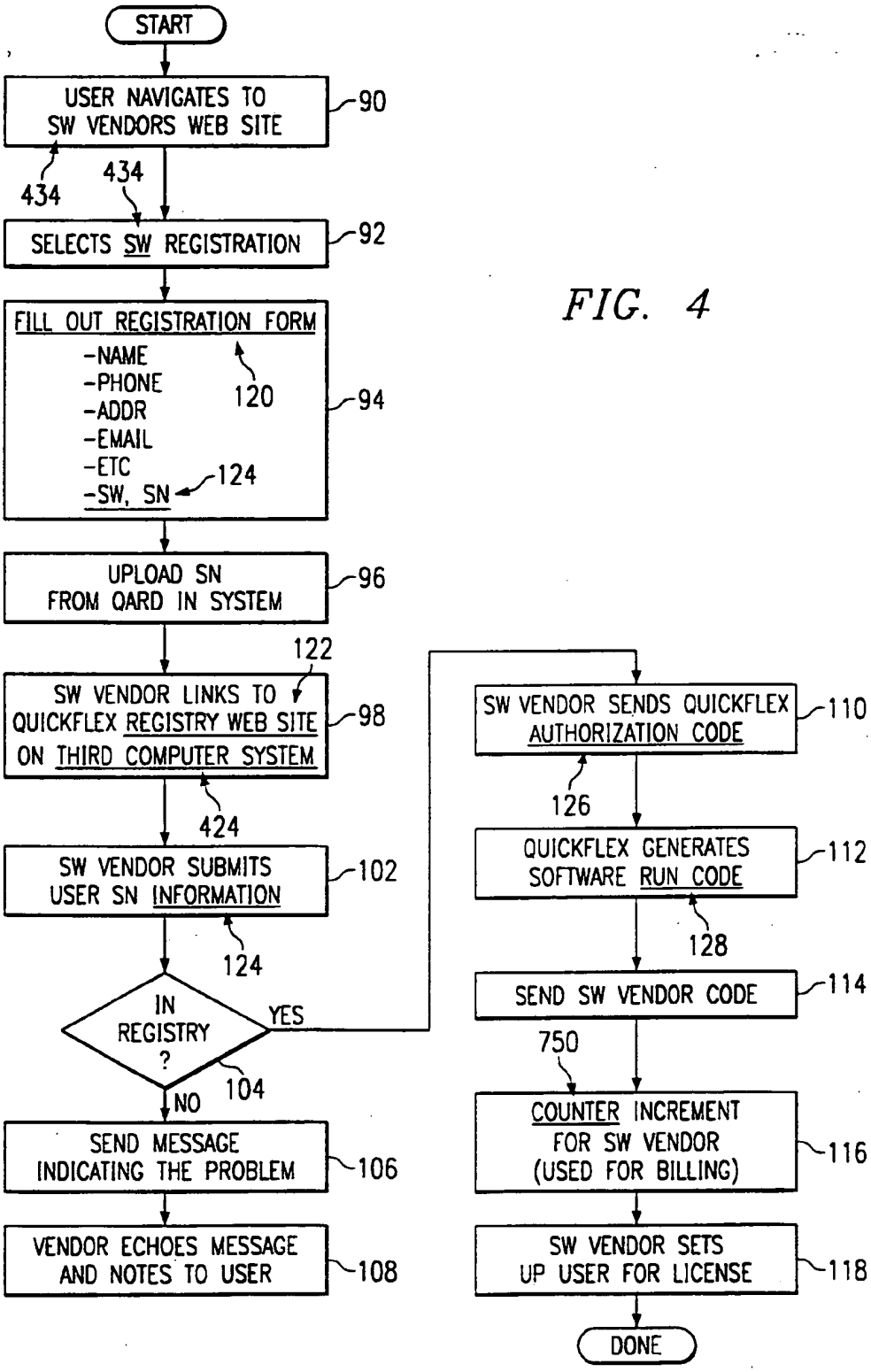
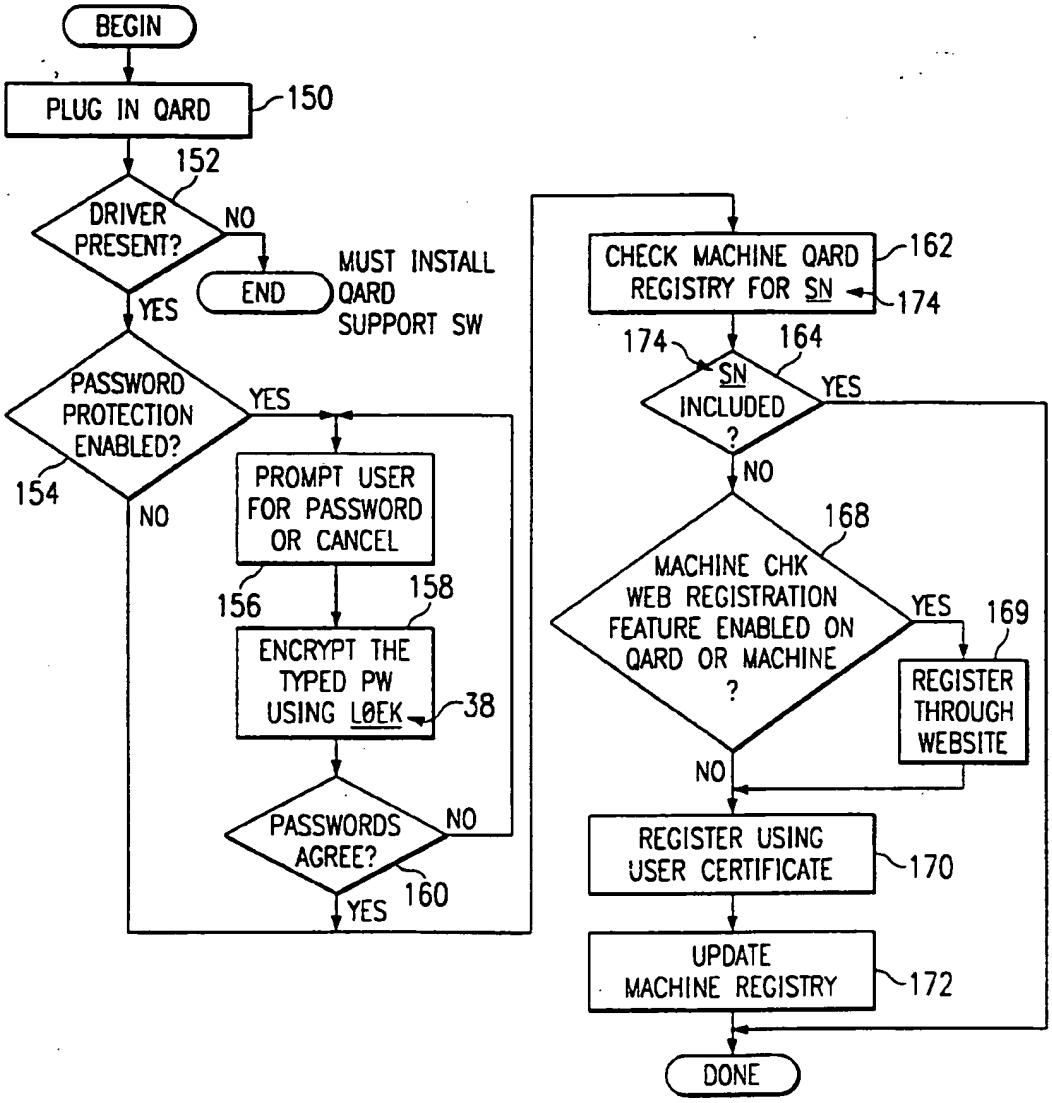

START
USER NAVIGATES TO SW VENDORS WEB SITE
90
434
434
SELECTS SW REGISTRATION
92
FILL OUT REGISTRATION FORM
-NAME
-PHONE
-ADDR
-EMAIL
-ETC
-SW, SN
120
124
94
UPLOAD SN FROM QARD IN SYSTEM
96
122
SW VENDOR LINKS TO QUICKFLEX REGISTRY WEB SITE ON THIRD COMPUTER SYSTEM
98
424
SW VENDOR SUBMITS USER SN INFORMATION
102
124
IN REGISTRY ?
YES
NO
104
SEND MESSAGE INDICATING THE PROBLEM
106
VENDOR ECHOES MESSAGE AND NOTES TO USER
108
SW VENDOR SENDS QUICKFLEX AUTHORIZATION CODE
110
126
QUICKFLEX GENERATES SOFTWARE RUN CODE
112
128
SEND SW VENDOR CODE
114
750
COUNTER INCREMENT FOR SW VENDOR (USED FOR BILLING)
116
SW VENDOR SETS UP USER FOR LICENSE
118
DONE

FIG. 4

| Step: | Description: 90 |
|---|---|
| 1<br>34 | If QUICKFLEX support software is not present on the machine, navigate online to QUICKFLEX web site and download the driver support software for Quick Cards. Execute and install the support software. |
| 2 | Insert the Card into the machine. |
| 3 | Driver checks if Card is Level 0 password protected. 130 |
| 4 | If required, user is prompted for a password before proceeding. |
| 5 | Driver checks if Card SN is in the local Card registry list (kept on machine). |
| 6 | If the Card is not included in the list, user is told that the Card must be registered for use on the machine and asks the user if they want to proceed with machine registration. Otherwise the Card is not useable. |
| 7 | User enters registration information, just as if registering the Card for the 1st time. |
| 8 | Information is checked against the QUICKFLEX registry database. |
| 9 | Machine registry file is updated to include the Card so it is useable on the machine. |
| 10 | LOEK for the Card is downloaded to the machine for inclusion in the registry. |
| 11 | Software using the Card protection must be installed and re-registered for the machine. Software vendors could allow the authorization file from another machine for the Card to be copied on other machines for use with the same Card if desired. |

*FIG. 5*

FIG. 6




| Step: | Description: |
|---|---|
| 1 | User indicates desire to password protect or change the password protection of the Qard to the utility program. |
| 2 | Program prompts user for current password if currently password protected. Then encrypts it using LOEK, checking it against the programmed password on Qard. |
| 3 | User is prompted for a new password and a reentry of it to insure correctness. |
| 4 | New password is encrypted using LOEK and the encrypted password is programmed into the Qard. |

FIG. 7

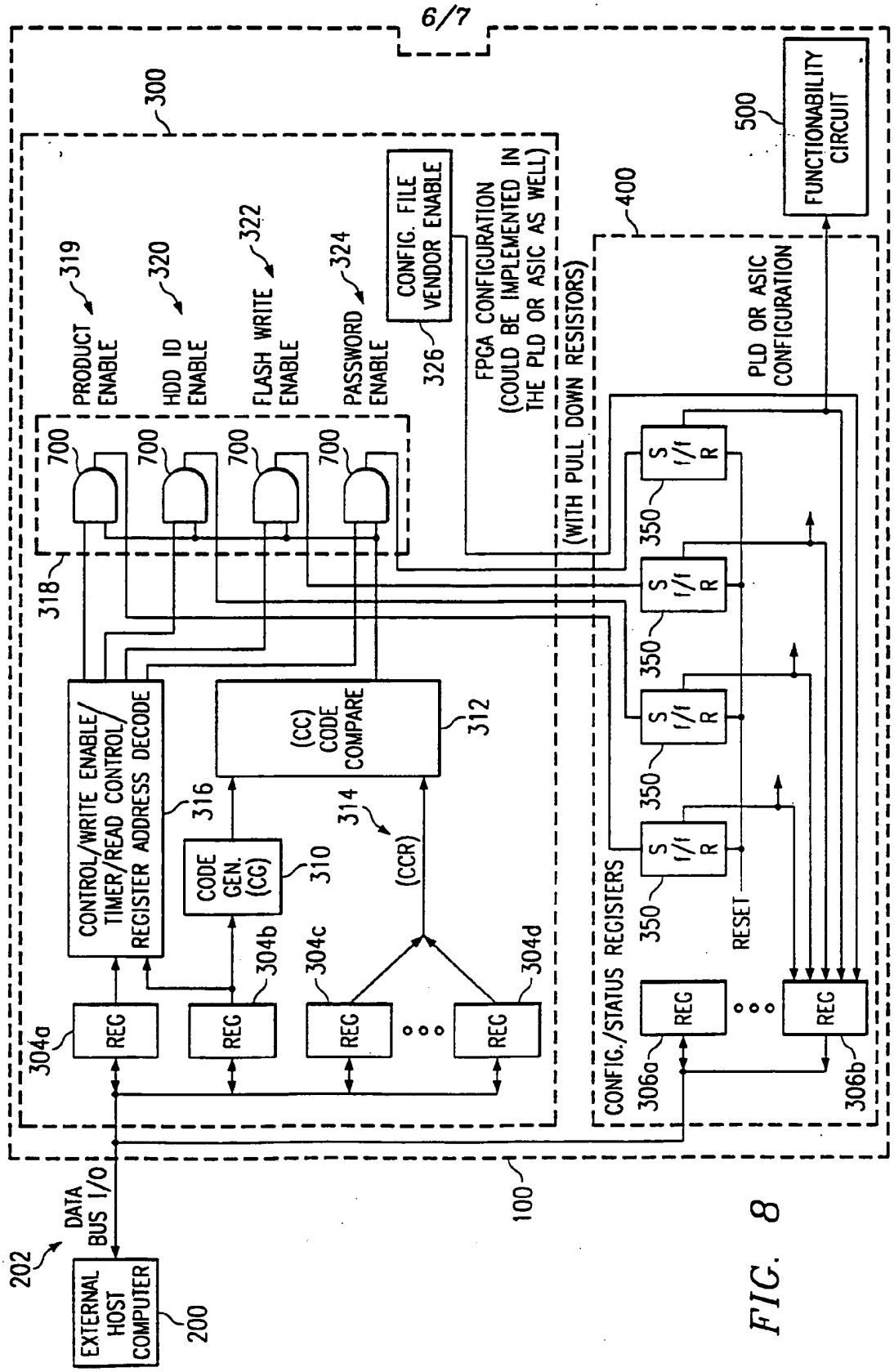

300
PRODUCT ENABLE
319
HDD ID ENABLE
320
FLASH WRITE ENABLE
322
PASSWORD ENABLE
324
CONFIG. FILE VENDOR ENABLE
326
FPGA CONFIGURATION (COULD BE IMPLEMENTED IN THE PLD OR ASIC AS WELL)
(WITH PULL DOWN RESISTORS)
400
500
FUNCTIONABILITY CIRCUIT
PLD OR ASIC CONFIGURATION
700
318
CONTROL/WRITE ENABLE/ TIMER/READ CONTROL/ REGISTER ADDRESS DECODE
316
(CC) CODE COMPARE
312
CODE GEN. (CG)
310
314
(CCR)
304a
304b
304c
304d
REG
CONFIG./STATUS REGISTERS
350
S
1/1
R
RESET
306a
306b
100
DATA BUS I/O
202
EXTERNAL HOST COMPUTER
200

FIG. 8

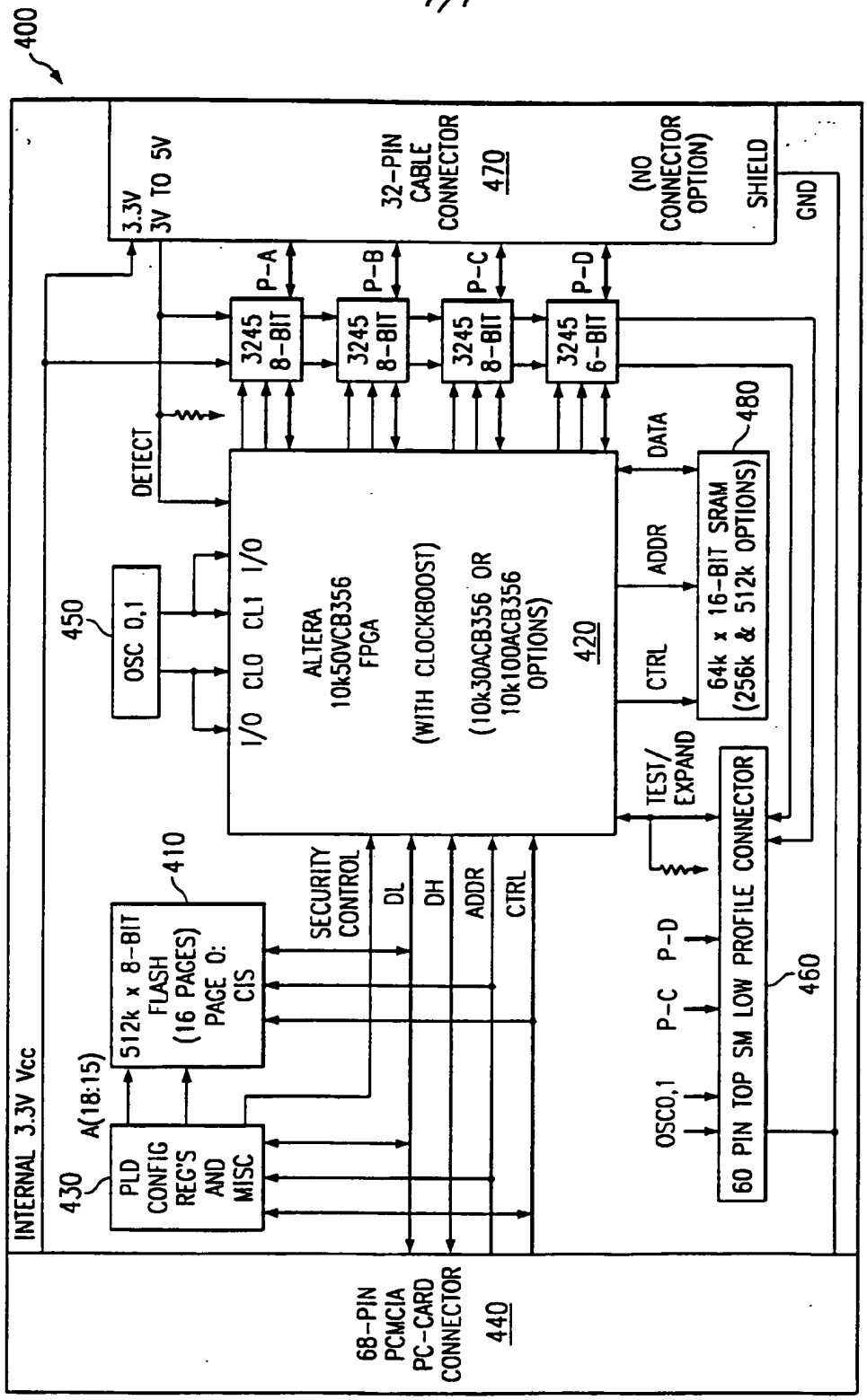

400
3.3V
3V TO 5V
32-PIN
CABLE
CONNECTOR
470
(NO
CONNECTOR
OPTION)
SHIELD
GND
P-A
P-B
P-C
P-D
3245
8-BIT
3245
8-BIT
3245
8-BIT
3245
6-BIT
DETECT
DATA
480
ADDR
CTRL
64k x 16-BIT SRAM
(256k & 512k OPTIONS)
450
OSC 0,1
I/O CLO CL1 I/O
ALTERA
10k50VCB356
FPGA
(WITH CLOCKBOOST)
(10k30ACB356 OR
10k100ACB356
OPTIONS)
420
I/O
TEST/
EXPAND
410
512k x 8-BIT
FLASH
(16 PAGES)
PAGE 0:
CIS
SECURITY
CONTROL
DL
DH
ADDR
CTRL
P-D
P-C
OSC0,1
60 PIN TOP SM LOW PROFILE CONNECTOR
460
INTERNAL 3.3V Vcc
A(18:15)
430
PLD
CONFIG
REG'S
AND
MISC
68-PIN
PCMCIA
PC-CARD
CONNECTOR
440

FIG. 9